REMARKS

This Amendment is filed in response to the Final Office Action mailed March 18, 2010 in connection with a Request for Continued Examination and a Petition for a 1-Month Extension of Time. The Applicant respectfully requests reconsideration. All objections and rejections are respectfully traversed.

Claims 1-11 and 12-34 are pending in the application.

Claims 1, 5, 17, 24, and 27 have been amended.

No new claims have been added.

Claim Objection

At paragraph 2 of the Final Office Action, claim 17 was objected to in relation to a typographical error. The Applicant has corrected this typo.

Claim Rejections - 35 U.S.C. §112, second paragraph

At paragraph 3 of the Final Office Action, claim 27 was rejected under 35 U.S.C. §112, second paragraph in relation to an alleged omission of an essential step.

The applicant has amended claim 27 and accordingly believes this rejection to be moot.

Claim Rejections - 35 U.S.C. §102

At paragraphs 4-12 of the Final Office Action, claims 1-3, 13, 20, 22-24 and 28-30 were rejected under 35 U.S.C. §102(b) over Karol et al., U.S. Patent No. 6,122,275 (hereinafter "Karol").

Claims 1-3, 13 and 20:

The Applicant's claim 1, representative in part of the claims 1-3, 13 and 20, recites (emphasis added):

> 1. A method for modifying packet header data transferred from a context memory internal to a forwarding engine to an output buffer of the forwarding engine, the method comprising the steps of:

reading one or more instructions, by a processor of the forwarding engine, each instruction indicating an operation to modify the packet header data;

generating, in response to the one or more instructions, one or more commands wherein each command is associated with the operation to modify the packet header data;

placing the one or more commands in a data structure;

initiating a transfer of the packet header data from the context memory internal to the forwarding engine to the output buffer of the forwarding engine; and

performing, by a data mover in the forwarding engine coupled to the context memory and the output buffer and operating independently from the processor, ***the operations associated with the one or more commands*** contained in the data structure, to modify the packet header data as directed by the one or more commands ***while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine.***

Karol describes a switch where "arithmetic logic unit (ALU) functionality is added to [a] header processing circuitry." *See* col. 2, lines 5-8. Karol describes that an incoming cell (Fig. 1, 110) to the switch may include an "operand field 112" and/or an operand may be obtained from an operand state field 121 in a lookup table 120. Based on the operand field 112 of the incoming cell 110 and/or the operand state field 121 in the lookup table 120, "a computation is performed in ALU 130 and operand state 121 or operand field 142 in outgoing cell 140, or both, are updated with the result." *See* col. 2, lines 62-65. An opcode that describes the computation to be performed by the ALU 130 is supplied by opcode control circuitry 160. *See* col. 2, lines 15-22. Karol describes that his switch may operate fast enough so that a "combine function" can be "executed in real time", keeping pace with the throughput rate of the switch. *See* col. 7, lines 46-55 and col. 1, lines 12-28.

The Applicant respectfully urges that Karol is silent concerning the Applicant's claimed "***initiating a transfer of the packet header data from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***" and "***performing … the operations associated with the one or more commands … while the***

packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine."

Karol lacks any mention of a **context memory internal to a forwarding engine** and an **output buffer of the forwarding engine**, and accordingly may not fairly be interpreted as teaching that packet header data should be modified **while it is being transferred between these two structures, as opposed to, for example, prior to a transfer being initiated**.

Karol may read an operand for his ALU from an incoming cell (*see* Fig. 1, 110). However, a mere cell may not fairly be considered a **context memory internal to a forwarding engine.** A cell is not a hardware structure. Further, Karol may update an outgoing cell (140) after the ALU (130) performs the computation. However, a mere cell may not fairly be considered **an output buffer of the forwarding engine.** Further, Karol describes that he may read an operand for his ALU (130) from a lookup table (120) and potentially later update the lookup table (120) after the ALU (130) performs the computation. However, a single lookup table (120) may not fairly be considered to be **both a context memory internal to a forwarding engine** and **an output buffer of the forwarding engine.** One lookup table (120) may not be fairly interpreted as both these separate structures.

The Applicant respectfully directs the Examiner's attention to Fig. 3 of the present application, which shows an example which may be helpful in understanding the Applicant's claims. As shown in Fig. 3, an example forwarding engine 300 includes one or more context memories 430 and an output buffer 700. A transfer (note arrow connecting 430 to 700) of packet header data is initiated from a context memory 430 internal to the forwarding engine 300 to the output buffer 700 of the forwarding engine 300. While the packet header data is in transit from the context memory 430 internal to the forwarding engine 300 to the output buffer 700 of the forwarding engine 300, a data mover (*see* Applicant's Fig. 4, 470) performs operations on the transferring packet header data.

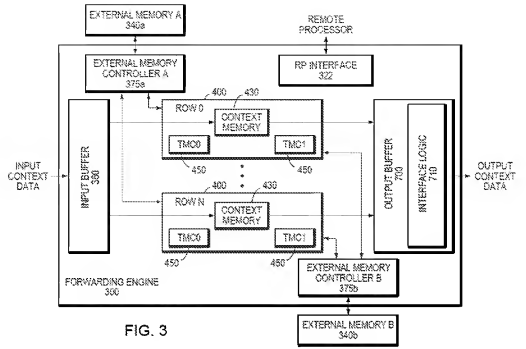

FIG. 3

The Applicant respectfully urges that Karol does not suggest this type of operation.

The Office Action points to general discussion in Karol at col. 7, lines 46-55 of executing a function "in real time." However, this in no way suggests packet header data should be modified **while it is being transferred between a context memory internal to a forwarding engine** and **an output buffer of the forwarding engine**. Karol simply uses the term "real time" to refer to processing sufficiently fast to keep pace with the throughput rate of his switch. For example, if a switch supports 100 Mb/s throughput, Karol considers "real-time" processing to be processing that can keep up with that throughput rate. Karol makes this clear at col. 1, lines 12-28, among other places, stating that "[i]ntelligent switches, on the other hand, suffer from technological limitations associated with being software based, and therefore cannot operate in real time, except for lower-throughput switches. (Routers, most of which operate below 100 Mb/s, are one example of intelligent switches.)" Thus, Karol simply envisions "real-time" processing to be processing that is sufficiently fast to keep pace with the throughput rate of the switch, and does not imply by the term that data is changed while it is being transferred between a context memory internal to a forwarding engine and an output buffer of the forwarding engine.

Accordingly, the Applicant respectfully urges that Karol is legally insufficient to anticipate the present claims under 35 U.S.C. §102 because of the absence of the Applicant's claimed novel "***initiating a transfer of the packet header data from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***" and "***performing … the operations associated with the one or more commands … while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine.***"

Claims 22-24 and 28-30:

The Applicant's claim 28, representative in part of the claims 22-24 and 28-30, recites (emphasis added):

> 28. A computer readable medium storing computer executable instructions for execution in a processor for:
>
> reading one or more instructions indicating an operation to modify packet header data;
>
> generating, in response to the one or more instructions, one or more commands wherein each command is associated with the operation to modify the packet header data;
>
> placing the one or more commands in a data structure;
>
> ***holding the one or more commands and not performing the operations associated with the one or more commands until initiation of a transfer of packet header data from a context memory internal to a forwarding engine to an output buffer of the forwarding engine***; and
>
> performing the operations associated with the one or more commands contained in the data structure, to modify the packet header data as directed by the one or more commands while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine

Karol is described above.

The Applicant respectfully urges that Karol is silent concerning the Applicant's claimed "***holding the one or more commands and not performing the operations associated with the one or more commands until initiation of a transfer of packet header data….***"

Karol lacks any mention of **holding one or more commands and not performing the operations associated with the one or more commands until initiation of a transfer of packet header data**, as opposed to, for example, performing the operations associated with the commands before any transfer of packet header data is begun. Karol merely suggests, at col. 7, lines 46-55, executing a function "in real time." As explained above, Karol simply uses the term "real time" to refer to processing that is sufficiently fast to keep pace with the throughput rate of the switch. For example, if switch supports 100 Mb/s throughput, "real-time" processing is processing that can keep up with that throughput rate. *See* Karol col. 1, lines 12-28. Karol does not suggest that a system should hold off performing operations until a transfer is initiated, and perform the operations as part of the transfer, as opposed to, for example,before any transfer of packet header data is begun.

Accordingly, the Applicant respectfully urges that Karol is legally insufficient to anticipate the present claims under 35 U.S.C. §102 because of the absence of the Applicant's claimed novel "***holding the one or more commands and not performing the operations associated with the one or more commands until initiation of a transfer of packet header data....***"

Claim Rejections - 35 U.S.C. §103

At paragraphs 14-24 of the Final Office Action, claims 4, 7, 9, 10, 14, 15, 18, 21, 25 and 31 were rejected under 35 U.S.C. §103(a) over Karol in view of Henderson, U.S. Publication No. 2004/0042490 (hereinafter "Henderson").

At paragraphs 25-27 of the Final Office Action, claims 5, 16, 26 and 32 were rejected under 35 U.S.C. §103(a) over Karol in view of Henderson, in further view of Ueno, U.S. Publication No. 2002/0009050 (hereinafter "Ueno").

At paragraphs 28-30 of the Final Office Action, claims 8 and 19 were rejected under 35 U.S.C. §103(a) over Karol in view of Henderson, in further view of Deforche et al., U.S. Publication No. 2005/0232303 (hereinafter "Deforche").

The Applicant notes that claims 4, 5, 7-9, 10, 14-16, 18, 19, 21, 25, 26, 31 and 32 are dependent claims that depend from independent claims believed to be allowable for at

least the reasons discussed above. Claims 4, 5, 7-9, 10, 14-16, 18, 19, 21, 25, 26, 31 and 32 are believed to be allowable due to their dependency, as well as for other separate reasons.

In the event that the Examiner deems a telephone conversation desirable in disposition of this application, the Examiner is encouraged to call the undersigned attorney at (617) 951-2500.

In summary, all the independent claims are believed to be in condition for allowance and therefore all dependent claims that depend there from are believed to be in condition for allowance. The Applicant respectfully solicits favorable action.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/James A. Blanchette/
James A. Blanchette
Reg. No. 51,477
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927